

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа _____ информационных технологий и робототехники
Направление подготовки _____ информационные системы и технологии
Отделение школы (НОЦ) _____ информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы	
Разработка плагина Visual Studio для рефакторинга кода на основе платформы Roslyn	
УДК 004.416.6:004.31.02	

Студент

Группа	ФИО	Подпись	Дата
8И4Б	Петрова Ольга Анатольевна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Токарева О.С.	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст.преподаватель ОСГН	Хаперская А.В.			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД	Волков Ю.В.	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко И.В.	к.т.н.		

Планируемые результаты обучения

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, критерии АИОР
P1	Применять базовые и специальные естественнонаучные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.	Требования ФГОС (ОК-1, 3, 10, ПК-9, 12, 26), критерий 5 АИОР (п. 1.1)
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.	Требования ФГОС (ОК-6, ПК-2, 3, 5, 7, 10, 11, 13, 15, 17, 18), критерий 5 АИОР (п.1.1, 1.2)
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.	Требования ФГОС (ОК-1, ПК-1, 4, 5, 6), критерий 5 АИОР (п. 1.2)
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).	Требования ФГОС (ПК-1 - 14), критерий 5 АИОР (п. 1.3)
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.	Требования ФГОС (ПК-23 - 27), критерий 5 АИОР (п.1.4)
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.	Требования ФГОС (ОК-12, ПК-15 – 18, ПК-29 – 37), критерий 5 АИОР (п. 1.5)
Универсальные компетенции		
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.	Требования ФГОС (ОК-4, 6, ПК-7, 15, 17), критерий 5 АИОР (п. 2.1)
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком (углублённый английский язык), позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.	Требования ФГОС (ОК-11), критерий 5 АИОР (п. 2.2)
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций,	Требования ФГОС (ОК-2), критерий 5 АИОР (п. 2.3)
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.	Требования ФГОС (ОК-7, 8), критерий 5 АИОР (п. 2.4)
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.	Требования ФГОС (ОК-4, 5, 8, 9, 13), критерий 5 АИОР (п. 2.5)

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа _____ информационных технологий и робототехники
Направление подготовки _____ информационные системы и технологии
Отделение школы (НОЦ) _____ информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ Цапко И.В.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8ИЗБ	Петрова Ольга Анатольевна

Тема работы:

Разработка плагина Visual Studio для рефакторинга кода на основе платформы Roslyn

Утверждена приказом директора (дата, номер)	2092/с от 26.03.2018
---	----------------------

Срок сдачи студентом выполненной работы:	04.06.2018
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	<ul style="list-style-type: none">– Задание на выполнение выпускной квалификационной работы;– Программный код проектов компании ООО «ТомскАСУпроект».
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ul style="list-style-type: none">– Обзор предметной области;– Описание используемых технологий;– Проектирование инструмента рефакторинга;– Разработка инструмента рефакторинга;.

Перечень графического материала	Рисунки; диаграммы пакетов, классов, вариантов использования; алгоритмы удаления автокомментариев.
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Хаперская Алена Васильевна
Социальная ответственность	Волков Юрий Викторович
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Все разделы на русском языке	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	22.01.2018
--	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Токарева О.С.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И4Б	Петрова Ольга Анатольевна		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа _____ информационных технологий и робототехники
Направление подготовки _____ информационные системы и технологии
Уровень образования _____ бакалавриат
Отделение школы (НОЦ) _____ информационных технологий
Период выполнения _____ весенний семестр 2017/2018 учебного года

Форма представления работы:

Бакалаврская работа

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	04.06.18
--	----------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
20.03.18	Аналитический обзор (обзор технологий, выбор методов решения)	15
20.04.18	Проектирование	25
30.05.18	Реализация	40
23.05.18	Социальная ответственность	10
04.06.18	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Токарева О.С.	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко И.В.	К.Т.Н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8И4Б	Петрова Ольга Анатольевна

Инженерная школа	информационных технологий и робототехники	Отделение	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Человеческие ресурсы – 2 человека.
2. <i>Нормы и нормативы расходования ресурсов</i>	Количество календарных дней – 365; Количество рабочих дней – 247; Количество нерабочих дней – 118.
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды составляет 0,3.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	1. Описание целевой аудитории разрабатываемой системы. 2. Анализ конкурентных технических решений 3. Проведение QuaD-анализа.
2. <i>Планирование и формирование бюджета научных исследований</i>	1. Определение организационной структуры проекта. 2. Определение трудоемкости выполнения работ. 3. Разработка календарного плана выполнения проекта.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Расчет интегральных показателей эффективности исследования, выбор наилучшего исполнения.

Перечень графического материала:

1. <i>Оценка конкурентоспособности технических решений</i>
2. <i>Матрица SWOT</i>
3. <i>Альтернативы проведения НИ</i>

- | |
|--|
| 4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ |
|--|

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст.преподаватель ОСГН	Хаперская А.В.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И4Б	Петрова Ольга Анатольевна		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8И4Б	Петрова Ольга Анатольевна

Инженерная школа	Информационных технологий и робототехники	Отделение	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения.	1. Разработка плагина Visual Studio для рефакторинга кода на основе платформы Roslyn. Плагин предназначен для улучшения понятности кода и изменения его структуры, чтобы в будущем код было легче поддерживать и развивать.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Производственная безопасность 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности. 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.	1.1 Рассмотрены вредные факторы: – отклонение показателей микроклимата; – недостаточная освещенность рабочей зоны; – монотонный режим работы. 1.2 Рассмотрены опасные факторы: – электрический ток; – пожаровзрывобезопасность.
2. Экологическая безопасность: 2.1 Анализ воздействия объекта на окружающую среду; 2.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.	2.1 Рассмотрены негативно влияющие на экологию факторы при эксплуатации компьютера. 2.2 Решения по обеспечению экологической безопасности согласно нормативным документам.
3. Безопасность в чрезвычайных ситуациях: 3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения; 3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.	3.1 Перечень возможных ЧС, которые могут возникнуть при работе в помещении офиса. 3.2 Способы защиты от пожара и ликвидация последствий.

4. Правовые и организационные вопросы обеспечения безопасности: 4.1 Организационные мероприятия при компоновке рабочей зоны. 4.2 Специфика влияния продукта на рабочий процесс.	4.1 Организационные мероприятия по обеспечению безопасности трудящихся за персональным компьютером. 4.2 Влияние разработанной системы для автоматизации процесса управления жизненным циклом средств криптографической защиты информации.
--	--

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД	Волков Ю.В.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И4Б	Петрова Ольга Анатольевна		

РЕФЕРАТ

Выпускная квалификационная работа 82 с., 15 рис., 25 табл., 22 источника, 3 прил.

Ключевые слова: рефакторинг кода, анализ кода, редактирование кода, синтаксическое дерево, компилятор.

Цель работы – реализация инструмента рефакторинга кода на основе платформы Roslyn для организации ООО «ТомскАСУпроект».

В отчете приведены результаты разработки плагина Visual Studio для рефакторинга кода на основе платформы Roslyn.

Область применения: рефакторинг кода в среде разработки Visual Studio.

Практическая значимость работы: разработанный инструмент рефакторинга был применен к проектам организации ООО «ТомскАСУпроект» для удаления автокомментариев. Если в компании возникнет очередная потребность в рефакторинге кода, имеющего другие цели, то архитектура плагина позволяет расширять свой функционал.

Экономическая значимость работы: данный инструмент может быть использован для улучшения понятности кода и изменения его структуры, чтобы в дальнейшем код было легче поддерживать и развивать.

В работе использовались следующие программные продукты: среда разработки Microsoft Visual Studio 2017 и SDK-пакет .NET Compiler Platform. Программа реализована на языке программирования C#.

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В данной работе применены следующие термины с соответствующими определениями:

ПО – программное обеспечение.

API (application programming interface) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах.

.NET Framework – это интегрированный компонент Windows, который поддерживает создание и выполнение нового поколения приложений и веб-служб XML.

XML (eXtensible Markup Language) – расширяемый язык разметки.

XAML (eXtensible Application Markup Language) – расширяемый язык разметки для приложений, основан на XML для декларативного программирования приложений.

ASP.NET (Active Server Pages для .NET) – платформа разработки веб-приложений, в состав которой входит: веб-сервисы, программная инфраструктура, модель программирования.

MVC (model-view-controller) – схема использования нескольких шаблонов проектирования, применяемая для создания программных продуктов.

MSBuild – платформа сборки проекта.

NAnt – это свободное ПО для автоматизации процесса сборки ПО.

CLR (common language runtime) – исполняемая среда для выполнения программ, написанных на языках .NET.

CIL (common intermediate language) – «высокоуровневый ассемблер» виртуальной машины .NET. Промежуточный язык, разработанный фирмой Microsoft для платформы .NET Framework, является частью.

LINQ (language integrated query) – проект Microsoft по добавлению синтаксиса языка запросов, напоминающего SQL, в языки программирования платформы .NET Framework.

Компьютерная платформа – в общем смысле, это любая существующая среда выполнения, в которой должен выполняться вновь разрабатываемый фрагмент программного обеспечения или объектный модуль с учётом накладываемых этой средой ограничений и предоставляемых возможностей.

Оглавление

1	Анализ предметной области	18
1.1	Актуальность разработки инструмента рефакторинга программного кода.....	18
1.2	Обзор существующих инструментов рефакторинга	19
2	Выбор и описание технологий, используемых для реализации.....	21
2.1	Язык программирования C#	21
2.2	Среда разработки Microsoft Visual Studio	22
2.3	Платформа Roslyn	22
2.4	Выбор технологии реализации плагина Visual Studio	27
3	Проектирование.....	29
3.1	Проектирование плагина Visual Studio.....	29
3.2	Алгоритм удаления автокомментариев	30
3.2.1	Алгоритм получения списка XML комментариев	30
3.2.2	Алгоритм анализа происхождения комментария.....	31
3.2.3	Алгоритм удаления автокомментариев.....	32
3.3	Проектирование архитектуры приложения	33
4	Программная реализация.....	36
4.1	Реализация плагина Visual Studio.....	36
4.2	Реализация удаления автоматических комментариев.....	36
4.3	Результаты практического применения разработанного инструмента рефакторинга	38
5	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	40
5.1	Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	40

5.1.1	Потенциальные потребители результатов исследования ..	40
5.1.2.	Анализ конкурентных технических решений.....	41
5.1.3	Технология QuaD	43
5.1.4	SWOT-анализ.....	45
5.2	Определение возможных альтернатив проведения научных исследований	48
5.3	Планирование научно-исследовательских работ	49
5.3.1	Структура работ в рамках научного исследования	49
5.3.2	Определение трудоемкости выполнения работ.....	50
5.3.3	Разработка графика проведения научного исследования....	51
5.3.4	Бюджет научно-технического исследования (НТИ)	56
5.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .	61
6	Социальная ответственность	64
6.1	Производственная безопасность	64
6.1.1	Вредные производственные факторы.....	65
6.1.1.1	Отклонение показателей микроклимата	65
6.1.2.	Опасные производственные факторы.....	70
6.2	Экологическая безопасность	72
6.3	Безопасность в чрезвычайных ситуациях	72
6.4	Правовые и организационные вопросы обеспечения безопасности	73
6.4.1	Специальные правовые нормы трудового законодательства	73
6.4.2	Организационные мероприятия при компоновке рабочей зоны.....	74

Заключение	76
Список использованных источников	77
Приложение А.	80
Приложение Б	81
Приложение В.....	82

Введение

Рефакторинг кода представляет собой процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения [1]. Рефакторинг изначально не предназначен для исправления ошибок и добавления новой функциональности. Его цель – сделать код программы более легким для понимания, чтобы в будущем код было легче поддерживать и развивать.

В компании ООО «ТомскАСУпроект», основной деятельностью которого является разработка ПО, ко многим проектам был применен сторонний инструмент автоматической генерации комментариев, который в результате только загромодил код комментариями, не несущих за собой смысловой нагрузки. В данный момент для рефакторинга в организации используется плагин ReSharper, но он обладает недостаточным функционалом для решения данной проблемы. Поэтому было предложено разработать новое собственное решение на основе платформы Roslyn, первой функцией которого будет удаление автоматических комментариев в программном коде.

Целью работы является реализация инструмента рефакторинга кода на основе Roslyn для организации ООО «ТомскАСУпроект».

Объектом исследования является программный код компании ООО «ТомскАСУпроект», содержащий автоматически сгенерированные комментарии.

В первом разделе приведены результаты анализа предметной области, включающие в себя актуальность данной работы и обзор существующих инструментов рефакторинга.

Во втором разделе представлено описание технологий, используемых для реализации инструмента рефакторинга, применение которых было требованием организации ООО «ТомскАСУпроект». Также в данном разделе

приведен обзор и обоснование выбора способа реализации плагина Visual Studio.

В третьем разделе представлены результаты проектирования программного приложения, включающего в себя: макет плагина Visual Studio, алгоритм удаления автокомментариев в программном коде и архитектуру приложения.

В четвертом разделе рассмотрена реализация плагина Visual Studio для рефакторинга программного кода, позволяющего удалять автоматические комментарии. Также приведены результаты практического применения разработанного программного продукта.

В пятом и шестом разделах изложены вопросы, связанные с финансовым менеджментом, ресурсоэффективностью, ресурсосбережением и социальной ответственностью проекта.

Практическая значимость. Разработанный инструмент рефакторинга был применен к проектам организации ООО «ТомскАСУпроект» для удаления автокомментариев. Если в компании возникнет очередная потребность в рефакторинге кода, имеющего другие цели, то архитектура плагина позволяет расширять свой функционал.

В работе использовались следующие программные продукты: среда разработки Microsoft Visual Studio 2017 и SDK-пакет .NET Compiler Platform. Программа реализована на языке программирования C#.

1 Анализ предметной области

1.1 Актуальность разработки инструмента рефакторинга программного кода

Комментарии, используемые при написании программного кода, являются немаловажной частью любого языка программирования, т.к. позволяют удобно пояснять различные участки кода. Они необходимы для того, чтобы разработчики, не писавшие данный участок кода, могли легко разобраться для чего он предназначен.

В языке C# имеется специальное средство, способное генерировать документацию в формате XML на основе специальных комментариев (далее XML комментарии) [2]. XML комментарии представляют собой однострочные комментарии, которые начинаются с трех слешей (///). Данные комментарии состоят из XML-дескрипторов, содержащих документацию по типам и членам типов, используемых в коде.

На основе XML документации можно составить графы зависимостей программных объектов, диаграммы классов и исходных кодов с гиперссылками и любую другую документацию. Очень важно, чтобы комментарии были осознанными и отображали корректную информацию.

На рисунке 1 представлен комментарий метода, сгенерированный автоматически сторонним продуктом. По данному комментарию невозможно понять, что делает этот метод, что обозначают принимаемые и возвращаемый параметры.

В данный момент на предприятии ООО «ТомскАСУпроект» около десятка проектов, каждый из которых включает сотни документов, содержат автоматически сгенерированные комментарии. Они не несут за собой смысловую нагрузку и только загромождают код.

```

/// <summary>
/// The call method.
/// </summary>
/// <param name="src">
/// The src.
/// </param>
/// <param name="method">
/// The method.
/// </param>
/// <param name="parameters">
/// The parameters.
/// </param>
/// <returns>
/// The <see cref="object"/>.
/// </returns>
ссылка: 0
public static object CallMethod(object src, MethodInfo method, object[] parameters)

```

Рисунок 1 – Комментарий метода, сгенерированный автоматически

Удаление автокомментариев вручную является долгой и монотонной работой. Поэтому возникла необходимость автоматизировать данный процесс. Данная задача относится к задаче рефакторинга кода, так как есть необходимость в редактировании кода, чтобы его можно было легко читать и поддерживать, при этом не изменяя его функциональность.

1.2 Обзор существующих инструментов рефакторинга

На данный момент популярными инструментами рефакторинга программного кода являются ReSharper и CodeRush.

ReSharper – плагин Microsoft Visual Studio, разработанный компанией JetBrains. Данный программный продукт предназначен для статического анализа кода, что позволяет произвести поиск ошибок в коде до компиляции. Предусматривает дополнительные средства подсветки синтаксиса, поиска, форматирования, навигации, генерации и оптимизации кода, предоставляет 40 автоматизированных рефакторингов, упрощает юнит-тестирование в средах MSTest и NUnit и др. Поддерживает языки программирования C# и VB.NET, а также предоставляет средства для работы с ASP.NET, ASP.NET MVC, XML, XAML, сценариями сборки NAnt и MSBuild [3].

CodeRush также является плагином Microsoft Visual Studio для рефакторинга и повышения производительности, разработанный компанией DevExpress.

CodeRush использует ситуативный статический анализ кода (выявление ошибок без необходимости компиляции), поддерживает дополнительные возможности для выявления и коррекции ошибок, автозавершение кода, навигацию, поиск, подсветку синтаксиса, форматирование, создание и оптимизацию кода, выполняет более 180 автоматических улучшений, упрощенное модульное тестирование при помощи NUnit, XUnit, MbUnit и MSTest [4].

Данные расширения являются платными, а также обладают недостаточным функционалом для организации ООО «ТомскАСУпроект». Например, они не предназначены для анализа комментариев в программном коде и не могут решить задачу удаления автокомментариев.

2 Выбор и описание технологий, используемых для реализации

В организации ООО «ТомскАСУпроект» разработка ПО преимущественно осуществляется на языке программирования С# в среде разработки Visual Studio. Требованием организации было использовать платформу Roslyn. В данном разделе приведено описание данных технологий.

Выбор технологии реализации плагина Visual Studio стоял за студентом. Обзор и обоснование выбора способа реализации плагина также приведены в данном разделе.

2.1 Язык программирования С#

С# – объектно-ориентированный язык программирования, разработанный в 1998—2001 годах компанией Microsoft. В настоящее время С# является одним из самых быстро развивающихся и востребованных языков. Он предназначен для разработки разнообразных приложений, которые выполняются в среде .NET Framework.

С# относится к языкам с С-подобным синтаксисом и является наиболее близким к Java и С++. Язык поддерживает полиморфизм, статическую типизацию, перегрузку операторов атрибуты, делегаты, свойства, события, обобщённые методы и типы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [5].

С# исключает некоторые проблемные модели своих предшественников. Например, С++ поддерживает множественное наследование классов, а в С# отказались от этого, но при этом допускается множественное наследование интерфейсов.

2.2 Среда разработки Microsoft Visual Studio

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows [6].

Visual Studio предоставляет удобный редактор исходного кода, который поддерживает технологию IntelliSense. Данная технология дает возможность простого рефакторинга кода, который может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя инструменты для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние плагины для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения.

2.3 Платформа Roslyn

Раньше компиляторы C# и Visual Basic представляли из себя чёрный ящик – на вход подавался исходный высокоуровневый код, а готовый CIL-код выдавался на выходе [7]. И что именно происходит во время данного преобразования было скрыто.

Компания Microsoft выпустила платформу для создания компиляторов, предоставляющую возможность передачи разработчикам всей

генерируемой компилятором информации. Данная платформа называется .NET Compiler Platform, более известная как Roslyn.

Roslyn – платформа с открытым исходным кодом, которая содержит в себе средства для анализа и разбора кода и компиляторы, которая написана и предназначена для языков программирования C# и Visual Basic. Средства анализа платформы Roslyn могут производить полный разбор кода, анализируя все поддерживаемые конструкции языка [8].

Анализ представляет собой разбор файлов, исходный код которых будет проверяться, а именно: получение синтаксического дерева и семантической модели.

Синтаксическое дерево является базовым элементом, необходимым для анализа кода. Именно по нему происходит перемещение в ходе анализа. Дерево строится на основе кода, приведенного в файле, следовательно каждый файл имеет свое синтаксическое дерево, которое является неизменяемым. Для изменения кода требуется новое синтаксическое дерево, которое может быть построено на основе первоначального [8].

Существует 3 основных элемента синтаксического дерева:

- Syntax nodes – одни из основных элементов дерева, представляют собой синтаксические конструкции языка. К этой категории относятся определения, объявления, операторы, выражения и т.п.;
- Syntax tokens – терминалы грамматики языка (лексемы, конечные символы), представляющие наименьшие синтаксические фрагменты кода. Они никогда не являются родителями других узлов. К этой категории относятся ключевые слова, идентификаторы, специальные символы и т.п.;
- Syntax trivia – дополнительная синтаксическая информация, которые в значительной степени несущественны для нормального понимания кода. Данные узлы не входят в конечную компиляцию кода. К этой категории относятся пробелы, переводы строк, комментарии, директивы препроцессора и т.п.

На рисунке 2 изображен пример синтаксического дерева, построенного по коду описания класса MyClass с пустым телом. Как видно из рисунка, небольшой код имеет довольно внушительное количество узлов: синий узел – определение класса типа syntax nodes; зеленые узлы – специальные слова (class), идентификаторы (MyClass) и фигурные скобки, относятся к типу syntax tokens; белые и серые узлы – пробелы и символы перевода строки, относятся к типу syntax trivia.

```
class MyClass
{
}
```

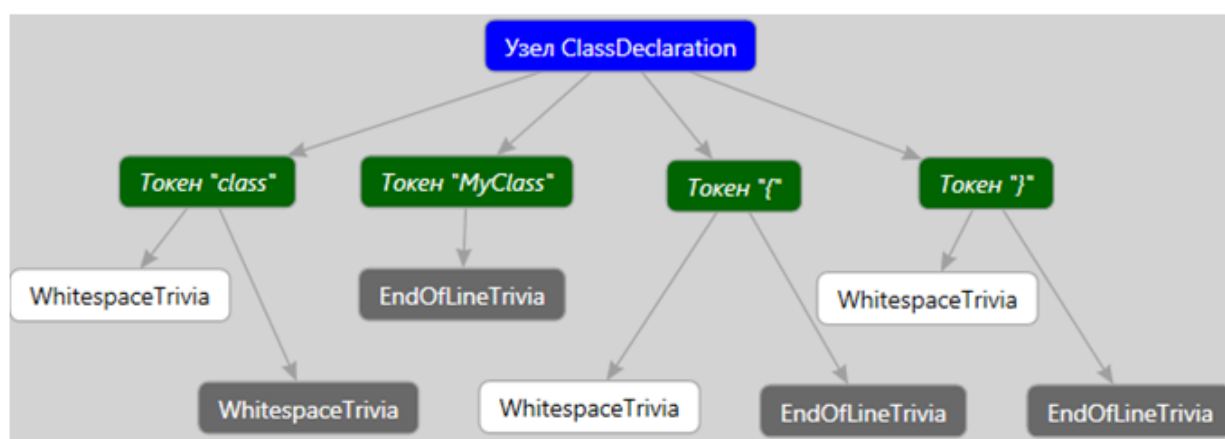


Рисунок 2 – Пример синтаксического дерева, построенного по коду описания класса MyClass с пустым телом

Обход синтаксического дерева представляет собой обход различных конструкций языка, разделенных по типам, которые соответствуют основным элементам дерева. Для каждого элемента существует свой метод обхода.

Синтаксические деревья представляют лексическую и синтаксическую структуру исходного кода. Хотя этой информации достаточно для описания всех деклараций и логики в коде, ее может быть недостаточно для определения того, на что ссылаются переменные и вызовы методов. Например, необходимо проанализировать следующий код:

```
a = 3;
```


Невозможно определить какого типа данная переменная, локальная ли она, является свойством или полем. Можно попробовать пройтись вверх по дереву до объявления переменной, но это было бы слишком расточительно с точки зрения производительности, более целесообразно использовать семантическую модель.

Семантическая модель – это очень мощный инструмент, предоставляющий всю семантическую информацию для одного исходного файла [9]. В данной работе применение семантической модели не используется, так как для анализа комментариев достаточно обхода синтаксические деревья.

Удобным инструментом отображения синтаксического дерева является расширение Visual Studio – Syntax visualizer, входящие в комплект Roslyn SDK. Данный инструмент облегчает понимание логики построения синтаксического дерева на практике. На рисунке 3 представлен пример отображения синтаксического дерева при помощи Syntax visualizer.

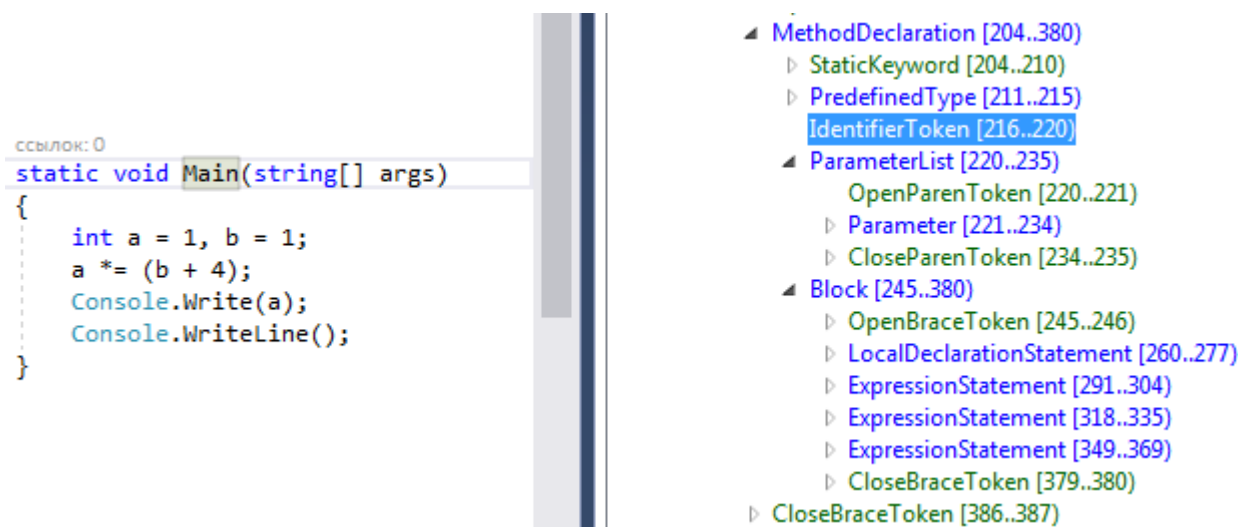


Рисунок 3 – Пример отображения синтаксического дерева при помощи
Syntax visualizer

Достоинства Roslyn:

- Открытый исходный код. Можно следить за процессом развития платформы, смотреть, что и как устроено. Если есть вопросы или

предложения, быстро связываться с разработчиками. А также можно менять исходный код под свои нужды, например реализовать анализатор кода для других языков программирования.

- Удобная и легкая навигация по дереву. Для перемещения по дереву или получения необходимых данных достаточно просто обращаться к свойствам узлов.

- Множество типов узлов. Можно подписаться на обход определённых узлов, соответствующих тем или иным конструкциям языка, тем самым анализируя интересующие нас участки. Кроме того, каждый тип узла предлагает характерный для него набор свойств, облегчая задачу получения необходимых данных.

- Семантическая модель. Сущность, позволяющая получать информацию об объектах и типах, предоставляя к тому же удобный интерфейс для этого, является очень сильной стороной платформы.

Недостатки Roslyn:

- Приходится изобретать собственные механизмы для, казалось бы, простых вещей, например метод `Equals` для сравнения узлов просто сравнивает ссылки на объекты, чего явно недостаточно.

- Открытие проектов может порождать различные проблемы. Порой Roslyn не может корректно открыть проект (не подцепляет какую-нибудь зависимость, файл и т.п.), из-за чего не удаётся получить корректную компиляцию и, как итог – семантическую модель.

- Программа, построенная на базе Roslyn, может потреблять много памяти (гигабайты). Для современных 64-битных компьютеров с большим объемом памяти это не является критичным, но учитывать эту особенность стоит.

2.4 Выбор технологии реализации плагина Visual Studio

Visual Studio также является открытой и расширяемой платформой. Например, если кто-то создает новый язык, разработчики могут добавить поддержку в Visual Studio для этого языка, в том числе возможность редактировать код, написанный на этом языке (как в виде текста, так и через графических дизайнеров), и скомпилировать код на этом языке, даже если Visual Studio не поддерживает язык по умолчанию.

Visual Studio предоставляет различные уровни расширяемости своего функционала, которые будут рассмотрены далее.

На самом базовом уровне можно использовать макросы, которые являются самым простым способом расширения и позволяют автоматизировать простые рутинные действия. Хотя макросы просты в использовании, они довольно ограничены в том, что можно выполнить, поскольку у них нет доступа к внутренней работе Visual Studio, например, они не могут создавать окна инструментов [10].

Подключаемый модуль (Add-In) – расширение, являющееся автономной сборкой, которая может автоматизировать задачи Visual Studio во многом так же, как это делают макросы. Однако, в отличие от макросов, может быть написана на любом языке и скомпилирована. Add-In имеет доступ через систему автоматизации к окнам инструмента и системе команд Visual Studio. У подключаемого модуля есть доступ к объекту DTE, который является корневым объектом, представляющим Visual Studio. Доступ к этому объекту позволяет манипулировать и управлять Visual Studio [11]. Например, объект DTE включает свойство ActiveDocument, которое возвращает ссылку на редактируемый документ.

Пакет VSPackages – следующий тип расширений, который, в отличие от подключаемых модулей, интегрируются прямо в IDE и загружаются как часть самой IDE. Таким образом, VSPackages имеют доступ ко многим

основным функциям Visual Studio и из-за глубокой интеграции VSPackages предлагает гораздо больше возможностей, чем Add-In.

На основе проведенного анализа для реализации плагина был выбран VSPackages.

3 Проектирование

3.1 Проектирование плагина Visual Studio

Плагин для Visual Studio будет реализован с помощью пакета VSPackage. Он будет представлять из себя пункт меню, при нажатии которого будут доступны функции инструмента рефакторинга.

В инструменте также должна быть заложена возможность дальнейшего наращивания функциональности по рефакторингу без глобальных изменений в архитектуре.

В рамках данной работы расширение будет обладать единственным функционалом – удалением автокомментариев в открытом решении.

Пользователю будет предоставлено 3 варианта по масштабу редактируемого кода:

- текущий документ – документ формата cs, который открыт в данный момент;
- текущий проект – проект, документ которого открыт в данный момент. Включает в себя все документы формата cs, относящиеся к проекту;
- все решение целиком – группа проектов, включенными в данное решение.

На рисунке 4 представлена диаграмма вариантов использования.

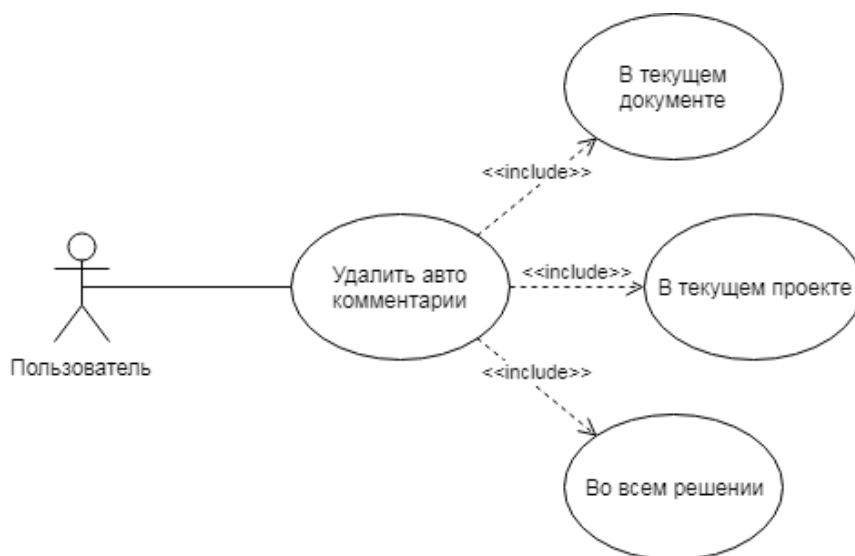


Рисунок 4 – Диаграмма вариантов использования

На рисунке 5 представлен макет плагина.

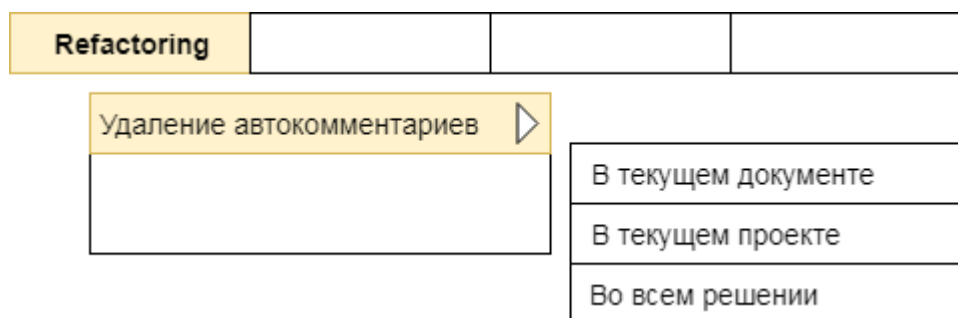


Рисунок 5 – Макет плагина

3.2 Алгоритм удаления автокомментариев

Процесс удаления автокомментариев можно разделить на 3 этапа:

- Получение списка XML комментариев;
- Анализ происхождения комментария (является ли он автоматическим);
- Редактирование файла (удаление из кода автокомментариев).

Данный алгоритм применяется для каждого документа в отдельности.

3.2.1 Алгоритм получения списка XML комментариев

Алгоритм получения списка XML комментариев из текста кода представлен на рисунке 6.

Входным параметром алгоритма является текст файла. По нему строится синтаксическое дерево. Из него необходимо извлечь корневой лист, чтобы получить доступ ко всем элементам синтаксического дерева. Далее из дочерних элементов корневого листа извлекаются список элементов типа *Syntax trivia*, так как к данному типу относятся комментарии (это описано в разделе «Платформа Roslyn»). Последним шагом, для получения списка XML комментариев, необходимо из списка элементов *Syntax trivia* вычленить элементы соответствующие типу XML комментариев.



Рисунок 6 – Алгоритм получения списка XML комментариев

Далее для каждого комментария из списка будет проводится анализ на предмет его принадлежности к автокомментариям.

3.2.2 Алгоритм анализа происхождения комментария

В первую очередь необходимо выявить правила определения является ли комментарий автоматическим.

Был проведен анализ проектов компании, содержащих автокомментарии, и составлены таблица тегов комментариев в зависимости от его владельца (Приложение А). Тег `summary` используется во всех XML комментариев, тег `param` – в определении конструкторов, методов и делегатов, тег `returns` – только в определении методов.

Также была составлена таблица правил генерации XML комментариев (Приложение Б). Текст автокомментария строится с использованием имени владельца комментария. Почти для всех типов существует единое правило генерации, исключения составляют: свойства, конструкторы и интерфейсы. Таким образом, комментарий является автоматическим, если текст комментария соответствует выявленному правилу генерации текста комментария.

В приложении В представлен алгоритм анализа происхождения комментария. На вход алгоритма подается XML комментарий. На первом шаге определяется тип владельца комментария. Далее генерируется список элементов комментария по тегам (summary, param и returns). Для каждого элемента из списка необходимо определить его тип, имя владельца комментария и текст комментария. На основе имени владельца комментария генерируется текст автокомментария, который сравнивается с текстом существующего комментария, если они идентичны, то данный элемент комментария добавляется в список на удаление.

В результате анализа всех XML комментариев файла формируется список комментариев, которые являются автоматическими. Данные комментарии необходимо удалить путем редактирования файла.

3.2.3 Алгоритм удаления автокомментариев

На рисунке 7 представлен алгоритм удаления автокомментариев. На вход ему подается список листов на удаление. Первым шагом необходимо удалить данные листы из корневого элемента синтаксического дерева, полученного в первом алгоритме (пункт 3.2.1), для этого Roslyn предоставляет специальный метод. Далее необходимо получить новый текст кода из корневого элемента и изменить содержание файла на данный текст, т. е. перезаписать его.



Рисунок 7 – Алгоритм редактирования документа

В результате использования данного алгоритма код файла будет избавлен от автокомментариев.

3.3 Проектирование архитектуры приложения

Приложение состоит из трех модулей (диаграмма пакетов представлена на рисунке 8):

1. RefactoringPlugin представляет собой модель плагина, которая включает в себя класс, описывающий команды и подкоманды меню.
2. Refactoring – промежуточный пакет между плагином и анализатором. Включает в себя класс CodeWorker, реализующий обработку документов (методы GetSolution, GetProject, GetDocument) и получение синтаксического дерева с корневым элементом (методы GetTree и GetRoot). Данный пакет будет использоваться для последующих реализаций функционала плагина.

3. CodeAnalyzer предназначен для анализа и редактирования кода, включает следующие классы (диаграмма классов представлена на рисунке 9):

- `CommentStructure` – вспомогательный класс для определения является ли элемент комментария автоматическим. Именно из него берутся листы на удаление.
- `AutoCommentRules` – класс правил генерации автокомментариев. Метод `GenerateAuthoCommentText` генерирует измененное имя владельца, а остальные методы на основе измененного имени возвращают текст автоматического комментария.
- `AutoCommentAnalyzer` – главный класс, в котором реализован алгоритм анализа происхождения комментария. Содержит список элементов комментария типа `CommentStructure` и методы, предназначенные для анализа.

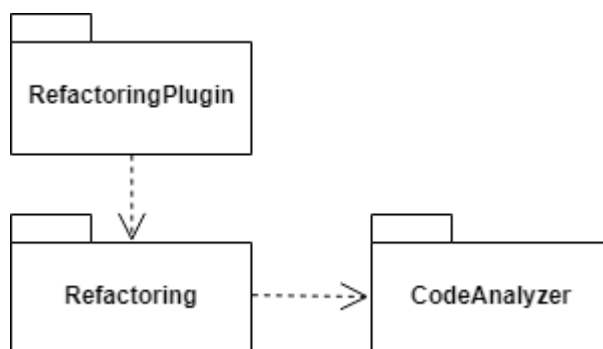


Рисунок 8 – Диаграмма пакетов

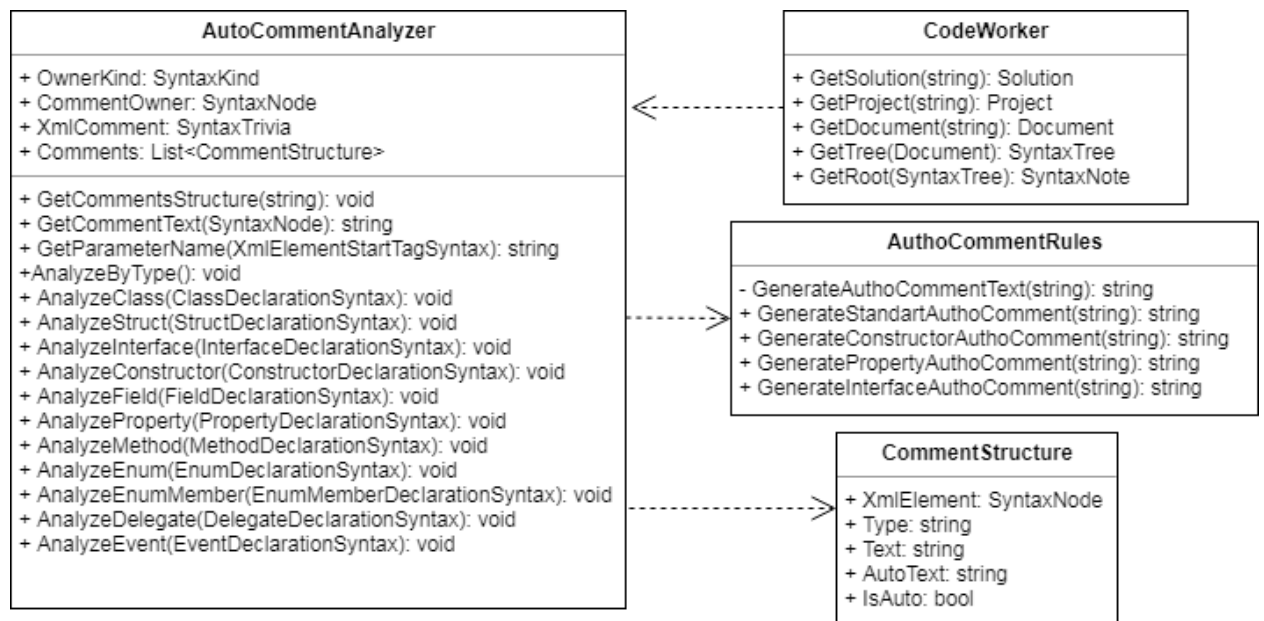


Рисунок 9 – Диаграмма классов

4 Программная реализация

4.1 Реализация плагина Visual Studio

Для создания нового пункта меню Visual Studio необходимо добавить проект шаблона VSIX и добавить в данный проект элемент Custom Command. В файле с расширением .vsct содержатся узлы <Menus> и <Symbols>, в которых необходимо перечислить названия новых команд меню и ID.

После сборки решения и запуска отладки будет отображаться экспериментальный экземпляр Visual Studio с новым пунктом меню.

Результатом сборки будет сгенерированный vsix файл, который является файлом установки расширений для Visual Studio. Запустив данный файл, сотрудники получают доступ к реализованным функциям рефакторинга.

4.2 Реализация удаления автоматических комментариев

Комментарии относятся к третьему основному типу элементов дерева – дополнительная синтаксическая информация (Syntax trivia), а XML комментарии к типу SingleLineDocumentationCommentTrivia. Именно по данному типу листов Syntax trivia будет получен список комментариев для дальнейшего анализа.

Каждый элемент SingleLineDocumentationCommentTrivia содержит список элементов типа XmlElement, каждый из которых содержит элемент типа XmlText, в котором содержится элемент XmlTextLiteralToken, хранящий в своих параметрах текст комментария. На рисунке 10 представлено расположение элемента XmlTextLiteralToken в дереве.

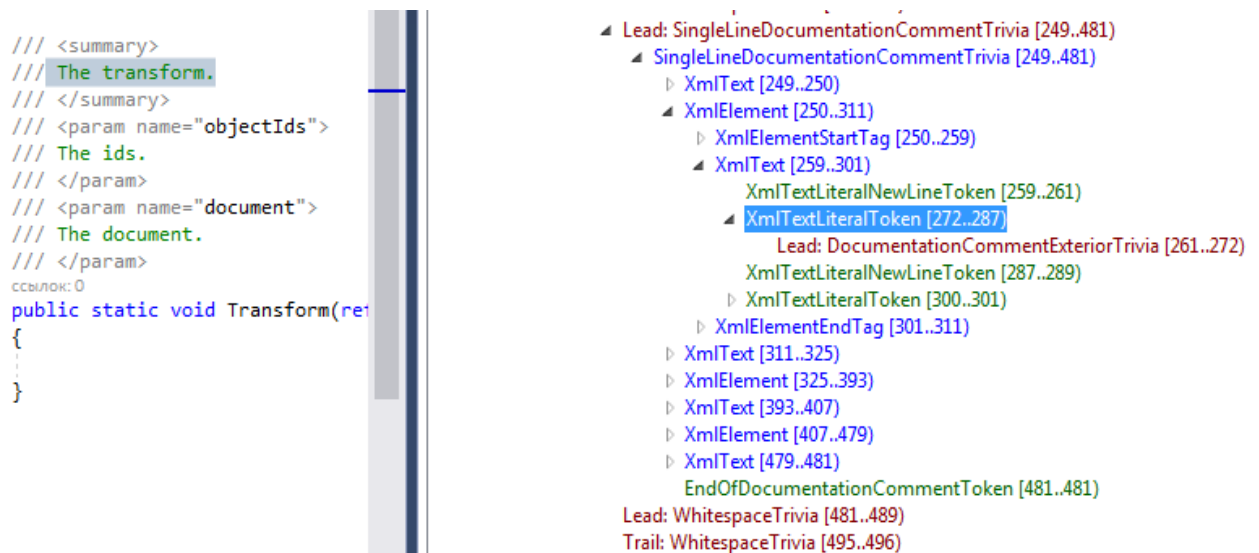


Рисунок 10 – Элемент синтаксического дерева, содержащий текст комментария

Каждому из владельцев комментария соответствует свой тип листа определения (например для класса – `ClassDeclarationSyntax`, для метода – `MethodDeclarationSyntax`, и т.д.). Данные классы являются наследником класса `NodeSyntax`, соответствующие типу листов `NodeSyntax` (пункт 2.3), но для получения имени владельца комментария необходимо использовать функционал дочерних элементов.

Например, для получения имени класса необходимо найти элемент IdentifierToken, так как в одном из его свойств содержится название класса. На рисунке 11 представлено расположение данного элемента в синтаксическом дереве.



Рисунок 11 – Элемент синтаксического дерева, содержащий имя
комментария

Чтобы удалить элементы автокомментариев из корневого листа, необходимо воспользоваться методом `RemoveNodes` (`IEnumerable<SyntaxNode> nodes, SyntaxRemoveOptions options`), где `nodes` – список листов на удаление, `options` – параметры удаления.

Для получения строкового представления листа предназначены 2 метода – `ToString()` и `ToFullString()`. Первый удаляет из текста все элементы `Syntax trivia`. Так как к данному типу относятся комментарии, то необходимо использовать второй метод.

4.3 Результаты практического применения разработанного инструмента рефакторинга

В результате реализации был получен работоспособный инструмент рефакторинга в виде плагина Visual Studio, позволяющий удалять автоматически сгенерированные комментарии. На рисунке 12 изображено меню плагина.

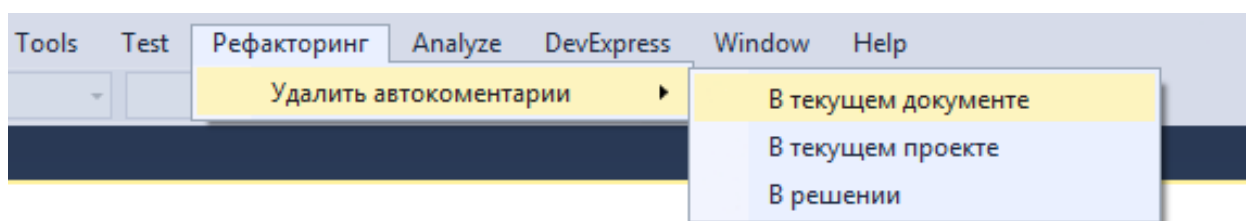


Рисунок 12 – Меню реализованного плагина

К проектам компании был применен инструмент рефакторинга для удаления автокомментариев. На рисунках 13 и 14 представлен код класса `Road` до и после применения инструмента рефакторинга соответственно. Удалилось более 50 % кода, представляющего собой автокомментарии, и читать его стало легче, так как теперь он не загроможден лишней информацией.

```

public class Road : IDisposable
{
    #region Fields

    /// <summary>
    /// The acad polyline.
    /// </summary>
    public readonly Polyline AcadPolyline;

    /// <summary>
    /// The is valid.
    /// </summary>
    public bool IsValid;

    /// <summary>
    /// The _all pickets.
    /// </summary>
    private readonly Dictionary<string, double> _allPickets = new Dictionary<string, double>();

    /// <summary>
    /// The _rub picket collection.
    /// </summary>
    private RubPicketCollection _rubPicketCollection;

    /// <summary>
    /// The _start picket.
    /// </summary>
    private string _startPicket = PicketWorker.StartPicket;

    #endregion

    #region Constructors and Destructors

    /// <summary>
    /// Initializes a new instance of the <see cref="Road"/> class.
    /// </summary>
    ссылка: 4
    public Road()

```

Рисунок 13 – Код класса Road до применения инструмента рефакторинга

```

public class Road : IDisposable
{
    #region Fields

    public readonly Polyline AcadPolyline;

    public bool IsValid;

    private readonly Dictionary<string, double> _allPickets = new Dictionary<string, double>();

    private RubPicketCollection _rubPicketCollection;

    private string _startPicket = PicketWorker.StartPicket;

    #endregion

    #region Constructors and Destructors

    ссылка: 4
    public Road()

```

Рисунок 14 – Код класса Road после применения инструмента рефакторинга

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью написания раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, технологий, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

Для этого необходимо оценить конкурентные решения с точки зрения эффективности, найти те характеристики, улучшение которых повысит конкурентоспособность инструмента и выбрать наиболее оптимальный путь выполнения работ с учетом показателей ресурсной, финансовой, бюджетной, социальной и экономической эффективности.

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1 Потенциальные потребители результатов исследования

В первую очередь необходимо рассмотреть целевой рынок и провести его сегментирование. Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар.

В зависимости от категории потребителей необходимо использовать соответствующие критерии сегментирования и выбрать два наиболее значимых для рынка. Один из критериев сегментации – вид потребителей, основанный на целевой аудитории. Вторым критерий – функционал инструмента рефакторинга кода.

В таблице 1 представлена карта сегментирования рынка на основе наиболее значимых критериев.

Таблица 1 – Карта сегментирования

		Целевая аудитория		
		Разработчик	Аналитик	Управляющий проектом
	Управление комментированием			
	Применение шаблонов для написания кода			
	Проектирование			

5.1.2. Анализ конкурентных технических решений

Рынок постоянно меняется, поэтому периодически необходимо проводить детальный анализ конкурирующих разработок. Данный анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим соперникам.

В данный момент на рынке существует следующие аналоги статического анализа и рефакторинга кода:

ReSharper (к1),

PVS-Studio (к2).

Критерии для сравнения и оценки ресурсоэффективности и ресурсосбережения подбираются, исходя из выбранных объектов сравнения с учетом их технических и экономических особенностей разработки, создания и эксплуатации.

Позиция разработки и конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot B_i ,$$

где К – конкурентоспособность научной разработки или конкурента;
 B_i – вес показателя (в долях единицы); B_i – балл i-го показателя.

Оценочная карта сравнения конкурентных технических решений представлена в таблице 2.

Таблица 2 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы			Конкуренто- способность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
Технические критерии оценки ресурсоэффективности							
1. Повышение производительности труда	0,15	5	5	5	0,75	0,75	0,75
2. Скорость работы	0,12	5	4	4	0,6	0,48	0,48
3. Удобство в эксплуатации	0,1	5	5	4	0,5	0,5	0,4
4. Потребность в ресурсах памяти	0,02	5	3	3	0,1	0,06	0,06
5. Функциональная мощность	0,1	3	5	5	0,3	0,5	0,5
6. Простота эксплуатации	0,08	5	4	4	0,4	0,32	0,32
7. Качество интеллек- туального интерфейса	0,05	5	4	5	0,25	0,2	0,25
Экономические критерии оценки эффективности							
8. Конкурентоспособность продукта	0,05	3	5	5	0,15	0,25	0,25
9. Уровень проникновения	0,08	2	5	5	0,16	0,4	0,4

на рынок							
10. Перспективность рынка	0,05	5	5	4	0,25	0,25	0,2
11. Цена	0,1	5	3	4	0,5	0,3	0,4
12. Послепродажное обслуживание	0,05	5	5	4	0,25	0,25	0,2
13. Финансовая эффективность научной разработки	0,05	5	4	4	0,25	0,2	0,2
Итого	1				4,46	4,46	4,41

На основе оценки конкурентоспособности можно сделать вывод о том, что уязвимостями разработки на фоне аналогов являются сравнительно невысокий функционал, так как разработка является узкоспециализированной, и имеет малый уровень проникновения на рынок в сравнении с другими существующими аналогами статического анализа и рефакторинга кода.

Конкурентными преимуществами разработки являются высокая скорость работы, простота эксплуатации, а также цена.

5.1.3 Технология QuaD

Технология QuaD – это гибкий инструмент измерения характеристик, которые описывают качество новой разработки и ее перспективность на рынке. Он позволяет принимать решение о целесообразности вложения денежных средств в научно-исследовательский проект.

В основе технологии QuaD лежит нахождение средневзвешенной величины групп показателей: показатели оценки коммерческого потенциала разработки и показатели оценки качества разработки.

Показатели оценки качества и перспективности новой разработки были подобраны исходя из выбранного объекта исследования с учетом его

технических и экономических особенностей разработки, создания и коммерциализации.

В соответствии с технологией QuaD каждый показатель оценивается по стобалльной шкале, где 1 – наиболее слабая позиция, а 100 – наиболее сильная. Веса показателей в сумме должны составить 1.

Оценка качества и перспективности по технологии QuaD определяется по формуле:

$$П_{ср} = \sum B_i \cdot Б_i ,$$

где $П_{ср}$ – средневзвешенное значение показателя качества и перспективности научной разработки; B_i – вес показателя (в долях единицы); $Б_i$ – средневзвешенное значение i -го показателя.

Оценочная карта конкурентных программных решений по технологии QuaD с учетом технических и экономических особенностей этой разработки приведена в таблице 3.

Таблица 3 – Оценочная карта QuaD для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы	Макс. балл	Относит. значение (3/4)	Средне-взвешенное значение (5x2)
Показатели оценки качества разработки					
1. Производительность труда	0,15	90	100	0,9	0,135
2. Скорость работы	0,12	100	100	1	0,12
3. Удобство в эксплуатации	0,1	80	100	0,8	0,08
4. Потребность в ресурсах памяти	0,02	90	100	0,9	0,018
5. Функциональная мощность	0,1	60	100	0,6	0,06

6. Простота эксплуатации	0,08	100	100	1	0,08
7. Качество интеллектуального интерфейса	0,05	90	100	0,9	0,045
Показатели оценки коммерческого потенциала разработки					
8. Конкурентоспособность продукта	0,05	75	100	0,75	0,0375
9. Уровень проникновения на рынок	0,08	30	100	0,3	0,024
10. Перспективность рынка	0,05	90	100	0,9	0,054
11. Цена	0,1	90	100	0,9	0,09
12. Послепродажное обслуживание	0,05	85	100	0,85	0,0425
13. Финансовая эффективность научной разработки	0,05	80	100	0,8	0,04
Итого	1				0,817

Полученное средневзвешенное значение показателя качества и перспективности разработки составляет 81,7% и позволяет считать ее перспективной, так как значение принадлежит промежутку от 80 до 100%.

5.1.4 SWOT-анализ

SWOT-анализ — метод стратегического планирования, который применяют для исследования внешней и внутренней среды проекта. Он проводится в несколько этапов.

Первый этап заключается в описании сильных и слабых сторон проекта, в выявлении возможностей и угроз для реализации проекта, которые проявились или могут появиться в его внешней среде. В таблице 4 приведены результаты анализа в виде матрицы с рекомендациями к дальнейшему развитию проекта

Таблица 4 – Матрица SWOT

	Сильные стороны: С1. Быстродействие. С2. Уникальный функционал. С3. Удобная и простая эксплуатация.	Слабые стороны: Сл1. Ограниченный функционал. Сл2. Узкий круг целевой аудитории. Сл3. Использование только для одного языка программирования. Сл4. Использование только для одной среды разработки
Возможности: В1. Расширение функционала. В2. Возможность работы системы для нескольких языков программирования.	Направления развития: 1. В1С1С2С4 – расширение функционала по требованиям заказчика. 2. В2С1С2С3 – разработка направлена на языки программирования: HTML, JS.	Сдерживающие факторы: 1. В1Сл1Сл2 – необходимость произвести анализ востребованных функций рефакторинга, а также временные ресурсы. 2. В2Сл3Сл4 – отсутствие инструментов реализации.
Угрозы:	Угрозы развития:	Уязвимости:

У1. Развитие и появление аналогов разрабатываемой системы.	1. У1С2 – появление схожего функционала у конкурентов, что приведет к снижению уникальности продукта.	1. У1Сл1Сл2Сл3Сл4 – конкурентный продукт пользуется большим спросом.
У2. Слабая заинтересованность продукта на рынке.	2. У2С1С3 – снижение мотивации к поддержке и улучшению продукта.	

Второй этап состоит в выявлении соответствия сильных и слабых сторон научно-исследовательского проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень необходимости проведения стратегических изменений.

Необходимо составить интерактивные матрицы проекта, в которых производится анализ соответствия параметров SWOT каждого с каждым (приведены на рисунках 5-8).

Каждый фактор помечается либо знаком «+» (означает сильное соответствие сильных сторон возможностям), либо знаком «-» (что означает слабое соответствие); «0» – если есть сомнения в том, что поставить.

Таблица 5 – Взаимосвязь сильных сторон проекта и возможностей

Возможности проекта		С1	С2	С3
	В1	0	+	+
	В2	+	+	+

Таблица 6 – Взаимосвязь слабых сторон проекта и возможностей

Возможности проекта		Сл1	Сл2	Сл3	Сл4
	В1	+	+	-	+
	В2	+	+	+	+

Таблица 7 – Взаимосвязь сильных сторон проекта и угроз

Угрозы		C1	C2	C3
	B1	0	+	+
	B2	+	+	+

Таблица 8 – Взаимосвязь слабых сторон проекта и угроз

Угрозы		Сл1	Сл2	Сл3	Сл4
	B1	+	0	-	-
	B2	+	+	+	+

5.2 Определение возможных альтернатив проведения научных исследований

Для выявления возможных альтернатив разработки проекта и доработки результатов был использован морфологический подход. Он основан на систематическом исследовании всех теоретически возможных вариантов, вытекающих из закономерностей строения объекта проектной деятельности. Синтез охватывает как известные, так и новые, необычные варианты, которые при простом переборе могли быть упущены. Путем комбинирования вариантов получают большое количество различных решений, ряд которых представляет практический интерес.

В таблице 9 в виде матрицы представлены возможные варианты реализации разработки.

Таблица 9 – Морфологическая матрица веб-приложения

	1	2	3
А. Количество языков программирования, для которых проводится анализ	1	2	Больше 2

Б. Платформа анализа кода	Roslyn	SonarQube	-
В. Вид программного решения	Плагин	Отдельное настольное приложение	Расширение анализатора среды разработки

Путём комбинации различных параметров морфологии проекта были определены три наиболее оптимальных варианта исполнения:

- A1B1B1;
- A2B3B3;
- A3B1B2.

Для данной матрицы наиболее оптимальным вариантом исполнения является первый.

5.3 Планирование научно-исследовательских работ

5.3.1 Структура работ в рамках научного исследования

При планировании комплекса предполагаемых работ необходимо определить структуры работ в рамках проекта, участников каждой работы и установить продолжительность работ. На основе этого строится график проведения проектной работы.

Для выполнения проектной работы были выявлены основные этапы работ и сформирована рабочая группа, в состав которой вошли руководитель и разработчик. Для каждого этапа работ установлен соответствующий исполнитель (таблица 10).

Таблица 10 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Составление ТЗ	1	Формулирование проблемы и	Руководитель

		разработка ТЗ	
	2	Обзор существующих решений в исследуемой области	Разработчик
	3	Исследование технологий, библиотек, фреймворков необходимых для разработки	Разработчик
	4	Утверждение выбранных технологий	Разработчик, руководитель
Планирование работ	5	Календарное планирование работ по теме	Разработчик, руководитель
	6	Составление правил генерации автокомментариев	Разработчик
	7	Проектирование системы	Разработчик
Реализация	8	Реализация инструмента, обладающего заявленными в ТЗ функциональными возможностями	Разработчик
Тестирование	9	Выявление ошибок в работе приложения и их исправление	Разработчик
Применение программного продукта	10	Применение функционала программного продукта к проектам организации	Разработчик, руководитель

5.3.2 Определение трудоемкости выполнения работ

Основную часть стоимости разработки в основном образуют трудовые затраты. Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Ожидаемое (среднее) значение трудоемкости $t_{ож}$ может быть определено по следующей формуле:

$$t_{ож\ i} = \frac{3t_{min\ i} + 2t_{max\ i}}{5},$$

где $t_{ож\ i}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.; $t_{min\ i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы чел.-дн.; $t_{max\ i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями:

$$T_{pi} = \frac{t_{ож\ i}}{Ч_i},$$

где T_{pi} – продолжительность одной работы; $t_{ож\ i}$ – ожидаемая трудоемкость выполнения одной работы; $Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе. Результаты расчетов приведены в таблице 11.

5.3.3 Разработка графика проведения научного исследования

Наиболее наглядным и простым для восприятия графиком проектных работ является ленточный график в форме диаграммы Ганта. Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни при помощи формулы:

$$T_{ki} = T_{pi} \cdot k_{кал},$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях; T_{pi} – продолжительность выполнения i -й работы в рабочих днях; $k_{кал}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})},$$

где $T_{\text{кал}}$ – количество календарных дней в году; $T_{\text{вых}}$ – количество выходных дней в году; $T_{\text{пр}}$ – количество праздничных дней в году.

В 2018 году 365 дней, из них выходных и праздничных – 118 дней. Таким образом, коэффициент календарности для 2018 года будет равен:

$$k_{\text{кал}} = \frac{365}{365 - 118} = \frac{365}{247} \approx 1,478.$$

Рассчитанные значения в календарных днях по каждой работе $T_{\text{ки}}$ округляются до целого числа.














Все рассчитанные значения приведены в таблице 11. Варианты использования взяты из раздела 2.

На основе полученных данных строится календарный план-график для максимального по длительности исполнения работ (рисунок 14).

Таблица 11 – Временные показатели проведения научного исследования

Название работы	Трудоемкость работ									Испол- нители	Длитель- ность работ в рабочих днях T_{pi}	Длитель- ность работ в календар- ных днях T_{ki}						
	$t_{min\ i},$ чел-дни			$t_{max\ i},$ чел-дни			$t_{ож\ i},$ чел-дни											
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3				Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Формулирование проблемы и разработка ТЗ	2	2	2	4	4	4	2,8	2,8	2,8	1	1	1	2,8	2,8	2,8	4	4	4
Обзор существующих решений в исследуемой области	2	2	2	4	4	4	2,8	2,8	2,8	1	1	1	2,8	2,8	2,8	4	4	4
Исследование технологий, библиотек, фреймворков необходимых для разработки	4	4	4	5	5	5	4,4	4,4	4,4	1	1	1	4,4	4,4	4,4	7	7	7
Утверждение выбранных	5	5	5	8	8	8	6,2	6,2	6,2	2	2	2	3,1	3,1	3,1	5	5	5

технологий																		
Календарное планирование работ по теме	2	2	2	4	4	4	2,8	2,8	2,8	2	2	2	1,4	1,4	1,4	2	2	2
Составление правил генерации автокомментариев	5	5	5	7	7	7	5,8	5,8	5,8	1	1	1	5,8	5,8	5,8	9	9	9
Проектирование системы	7	7	7	10	10	10	8,2	8,2	1,4	1	1	1	8,2	8,2	1,4	12	12	12
Реализация инструмента, обладающего заявленными в ТЗ функциональными возможностями	25	30	25	30	35	30	27	32	27	1	1	1	27	32	27	40	47	40
Выявление ошибок в работе приложения и их исправление	5	5	5	6	6	6	5,4	5,4	5,4	1	1	1	5,4	5,4	5,4	8	8	8
Применение функционала программного продукта к проектам организации	2	2	2	3	3	3	2,4	2,4	2,4	2	2	2	1,2	1,2	1,2	2	2	2
Итого:													62,1	67,1	62,1	92	99	92

Вид работ	Продолжительность выполнения работ										
	февраль	март			апрель			май			июнь
	3	1	2	3	1	2	3	1	2	3	1
Формулирование проблемы и разработка ТЗ											
Обзор существующих решений в исследуемой области											
Исследование технологий, библиотек, фреймворков необходимых для разработки											
Утверждение выбранных технологий			 								
Календарное планирование работ по теме			 								
Составление правил генерации автокомментариев											
Проектирование системы											
Реализация инструмента, обладающего заявленными в ТЗ функциональными возможностями											
Выявление ошибок в работе приложения и их исправление											
Применение функционала программного продукта к проектам организации											 



 Разработчик  Руководитель

Рисунок 15 – Календарный план-график проведения НИОКР по теме

5.3.4 Бюджет научно-технического исследования (НТИ)

5.3.4.1 Расчет материальных затрат НТИ

Данная статья включает стоимость всех материалов, используемых при разработке проекта.

В ходе разработки использовалось имеющееся оборудование, поэтому в материальные расходы внесены затраты на его амортизацию за 5 месяцев. Также в материальные расходы занесены затраты на приобретение лицензий для используемого программного обеспечения.

Материальные затраты, необходимые для данной разработки, представлены в таблице 12.

Таблица 12 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за единицу, тыс.руб			Затраты на материалы, (З _м), тыс.руб.		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Лицензия на среду разработки	шт.	1	1	1	78	78	78	78	78	78
Итого								78	78	78

5.3.4.2 Расчет затрат на специальное оборудование для научных (экспериментальных) работ

В данную статью включают все затраты, связанные с приобретением специального оборудования, необходимого для проведения работ по конкретной теме.

В ходе выполнения НТИ использовалось имеющееся компьютерное оборудование, поэтому его стоимость учитывается в калькуляции в виде амортизационных отчислений за 5 месяцев (таблица 13).

Таблица 13 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество			Цена за единицу, тыс.руб			Затраты на материалы, (З _м), тыс.руб.		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Амортизация оборудования	шт.	1	1	1	16	16	16	16	16	16
Итого								16	16	16

5.3.4.3 Основная заработная плата исполнителей темы

В настоящую статью включается основная заработная плата исполнителей, непосредственно участвующих в выполнении работ по данной теме.

Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20 –30 % от тарифа или оклада.

Статья включает основную заработную плату работников, непосредственно занятых выполнением НТИ и дополнительную заработную плату:

$$З_{зп} = З_{осн} + З_{доп} ,$$

где З_{осн} – основная заработная плата; З_{доп} – дополнительная заработная плата (20 % от З_{осн}).

Основная заработная плата рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot Т_p ,$$

где $Z_{\text{осн}}$ – основная заработная плата одного работника; T_p – продолжительность работ, выполняемых научно-техническим работником, раб.дней; $Z_{\text{дн}}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d},$$

где Z_m – месячный должностной оклад работника, руб.; M – количество месяцев работы без отпуска в течение года; F_d – действительный годовой фонд рабочего времени, раб. дн (таблица 12).

Месячный должностной оклад работника определяется по формуле:

$$Z_m = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_d) \cdot k_p$$

где $Z_{\text{тс}}$ – заработная плата по тарифной ставке, руб.; $k_{\text{пр}}$ – премиальный коэффициент, равный 0,3; k_d – коэффициент доплат и надбавок составляет примерно 0,2; k_p – районный коэффициент (для Томска – 1,3).

Расчёт основной заработной платы приведён в таблице 14.

Таблица 14 – Расчёт основной заработной платы

Исполнители	З _{тс} , руб	к _{пр}	к _д	к _р	З _м , руб	З _{дн} , руб	Т _р , дней	З _{осн} , руб
Руководитель от ТПУ	25.000				48.750	2.210	30	66.300
Руководитель от предприятия	35.000				68.250	3.095	30	92.850
Итого								159.150

5.3.4.4 Дополнительная заработная плата исполнителей темы

Расчет дополнительной заработной платы ведется по следующей формуле:

$$З_{\text{доп}} = k_{\text{доп}} \cdot З_{\text{осн}},$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12).

Расчёт основной заработной платы приведён в таблице 15.

Таблица 15 – Расчёт основной заработной платы

Исполнители	К _{доп}	З _{осн} , руб	З _{доп} , руб
Руководитель от ТПУ		66.300	7.956
Руководитель от предприятия		92.850	11.142
Итого			19.098

5.3.4.5 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования, пенсионного фонда и медицинского страхования от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}}),$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (составляет 0,3).

Расчет отчислений во внебюджетные фонды приведен в таблице 16.

Таблица 16 – Отчисления во внебюджетные фонды

Исполнители	З _{осн} , руб	З _{доп} , руб	к _{внеб}	З _{внеб} , руб
Руководитель от ТПУ	66.300	7.956		22.277
Руководитель от предприятия	92.850	11.142		31.198
Итого				53.475

5.3.4.6 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется по следующей формуле:

$$З_{\text{накл}} = \text{сумма статей (1 ÷ 5)} \cdot k_{\text{нр}},$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы. Величину коэффициента накладных расходов можно взять в размере 16%.

5.3.4.7 Формирование бюджета затрат научно-исследовательского проекта

По итогам расчётов, выполненных в предыдущих разделах, можно составить полный бюджет затрат на реализацию проекта (таблица 17). Так как затраты по соответствующим статьям для всех вариантов использования равны, в таблице приведены общие значения.

Таблица 17 – Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб	Примечание
1. Материальные затраты НТИ	78.000	Пункт 3.4.1
2. Затраты на специальное оборудование для научных работ	16.000	Пункт 3.4.2
3. Затраты по основной заработной плате исполнителей темы	159.150	Пункт 3.4.3
4. Затраты по дополнительной заработной плате исполнителей темы	19.098	Пункт 3.4.4
5. Отчисления во внебюджетные фонды	53.475	Пункт 3.4.5
6. Накладные расходы	52.116	16 % от суммы ст. 1-5
7. Бюджет затрат НТИ	377.839	Сумма ст. 1- 6

5.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Оценка целесообразности вариантов исполнения проекта выполняется с помощью интегрального показателя эффективности научного исследования, вычисляемой на основе финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат всех вариантов исполнения научного исследования.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}},$$

где $I_{\text{финр}}^{\text{исп}i}$ – интегральный финансовый показатель разработки; Φ_{pi} – стоимость i -го варианта исполнения; Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Т.к. стоимость всех вариантов использования одинакова, интегральные финансовые показатели также будут одинаковы и равны 1.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^n a_i \cdot b_i,$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки; a_i – весовой коэффициент i -го варианта исполнения разработки; b_i – балльная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания; n – число параметров сравнения.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 18. Критерии оцениваются по пятибалльной шкале.

Таблица 18 – Сравнительная оценка характеристик вариантов исполнения
проекта

Объект исследования Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Повышение производительности труда	0,15	5	5	5
2. Скорость работы	0,12	5	5	3
3. Удобство в эксплуатации	0,1	5	4	3
4. Потребность в ресурсах памяти	0,02	5	5	5
5. Функциональная мощность	0,1	4	5	5
6. Простота эксплуатации	0,08	5	4	4
7. Качество интеллектуального интерфейса	0,05	5	5	5
I_{pi}		3	2,92	2,58

Интегральный показатель эффективности вариантов исполнения разработки ($I_{испi}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{испi} = \frac{I_{pi}}{I_{финр}^{испi}}.$$

Так как интегральные финансовые показатели одинаковы и равны 1, то интегральные показатели эффективности вариантов исполнения разработки равны соответствующим интегральным показателям ресурсоэффективности ($I_{испi} = I_{pi}$).

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность

проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта ($\mathcal{E}_{\text{ср}}$):

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп}i}}{I_{\text{исп}1}}.$$

Результаты расчета сравнительной эффективности представлены в таблице 19.

Таблица 19 – Сравнительная эффективность разработки

№	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	1	1	1
2	Интегральный показатель ресурсо-эффективности разработки	3	2,92	2,58
3	Интегральный показатель эффективности	3	2,92	2,58
4	Сравнительная эффективность вариантов исполнения	1	0,973	0,86

Таким образом, можно сделать вывод, что самым эффективным исполнением с позиции ресурсоэффективности и финансовой эффективности является первое исполнение.

6 Социальная ответственность

Целью работы является реализация инструмента рефакторинга. Рефакторинг представляет собой процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения. Рефакторинг изначально не предназначен для исправления ошибок и добавления новой функциональности. Его цель – сделать код программы более легким для понимания, чтобы в будущем код было легче поддерживать и развивать.

Разработка программы велась исключительно при помощи компьютера. В данном разделе будут рассмотрены вопросы выполнения требований к безопасности и гигиене труда, к промышленной безопасности, охране окружающей среды и ресурсосбережению, непосредственно связанные с работой с ПК.

6.1 Производственная безопасность

В процессе трудовой деятельности на человека могут воздействовать опасные и вредные факторы.

Вредный производственный фактор — производственный фактор, воздействие которого на работника может привести к его заболеванию (например, опасность снижения зрения при недостаточном уровне освещения рабочего места) [12].

Опасный производственный фактор — производственный фактор, воздействие которого на работника может привести к его травме (например, опасность падения нахождения рабочего места на высоте) [12].

В таблице 20 представлены возможные вредные и опасные факторы при выполнении работ за ПЭВМ.

Таблица 20 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ

Источник фактора,	Факторы (по ГОСТ 12.0.003-74)	Нормативные документы
-------------------	-------------------------------	-----------------------

наименование видов работ	Вредные	Опасные	
1) Работа за ПЭВМ	1) Отклонение показателей микроклимата (температуры и влажности воздуха) 2) Недостаточная освещенность рабочей зоны; 3) Монотонный режим работы.	1) Опасность поражения электрическим током; 2) Опасность возникновения пожара.	1) СанПиН 2.2.4.548-96; 2) СанПиН 2.2.2/2.4.1340- 03; 3) СП 52.13330.2011; 4) ГОСТ Р 12.1.019- 2009 ССБТ; 5) СНиП 21-01- 97.

6.1.1 Вредные производственные факторы

6.1.1.1 Отклонение показателей микроклимата

Микроклимат производственных помещений – это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также температуры окружающих поверхностей [14].

Сочетание неблагоприятных параметров микроклимата может вызвать резкое ухудшение самочувствия, снижение производительности, а также привести к заболеваниям. Причиной этого будет являться перегрев или переохлаждение.

Высокая температура воздуха может вызвать перегрев организма, что приводит к быстрой утомляемости работающего. Низкая температура

воздуха может вызвать переохлаждение организма, что может привести к простудным заболеваниям.

Влажность воздуха оказывает влияние на терморегуляцию организма человека: низкая влажность может вызвать пересыхание слизистых оболочек работника, а высокая увеличивает теплопроводность воздуха, усиливает негативные факторы высокой и низкой температуры воздуха.

Подвижность воздуха также влияет на организм человека: способствует эффективной теплоотдаче при высоких температурах, а при низких может нанести вред.

Работа за ПЭВМ относится к категории Ia. К данной категории относятся работы с интенсивностью энерготрат до 120 ккал/ч (до 139 Вт), производимые сидя и сопровождающиеся незначительным физическим напряжением (ряд профессий на предприятиях точного приборо- и машиностроения, на часовом, швейном производствах, в сфере управления и т.п.) [14].

Оптимальные микроклиматические условия обеспечивают ощущение теплового комфорта в течении 8-часового рабочего дня с незначительным напряжением механизмов терморегуляции. На таблице 21 приведены оптимальные величины показателей микроклимата на рабочих местах категории Ia производственных помещений согласно СанПиН 2.2.4.548-96.

Таблица 21 – Оптимальные величины показателей микроклимата на рабочих местах категории Ia производственных помещений

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22 – 24	21 – 25	60 – 40	0,1
Теплый	23 – 25	22 – 26	60 – 40	0,1

Допустимые микроклиматические условия в отличие от оптимальных могут привести к ощущению теплового дискомфорта, ухудшению

самочувствия и понижению работоспособности. Они не вызывают нарушения состояния здоровья. Допустимые величины показателей микроклимата устанавливаются в случаях, когда по обоснованным причинам (по технологическим требованиям, техническим и экономическим) не могут быть обеспечены оптимальные величины. На таблице 22 приведены допустимые величины показателей микроклимата на рабочих местах категории Ia производственных помещений согласно СанПиН 2.2.4.548-96.

Таблица 22 – Допустимые величины показателей микроклимата на рабочих местах категории Ia производственных помещений

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	20 – 25	19 – 26	15 – 75	0,1
Теплый	21 – 28	22 – 29	15 – 75	0,1 – 0,2

Поддержание оптимальных параметров микроклимата поддерживается за счет: центрального отопления, вентиляции, проветривания помещения, влажной уборки, искусственного кондиционирования, увлажнителя воздуха.

6.1.1.2 Недостаточная освещенность рабочей зоны

Работа с компьютером подразумевает постоянный зрительный контакт с монитором. Низкий уровень освещенности рабочей зоны может привести к быстрому утомлению, головным болям, снижению остроты зрения и концентрации внимания, что может привести к ухудшению производительности труда.

В зависимости от источника света освещение подразделяется на естественное, искусственное и совмещенное. Помещения для эксплуатации ПЭВМ должны иметь совмещенное освещение.

В процессе разработки программных продуктов инженеру-программисту приходится различать объекты на мониторе. Данный вид работ относится к подразряду «Г» 3-го разряда зрительных работ (работы высокой точности).

Таблица 23 – Требования к освещению помещений промышленных предприятий для операторов ПЭВМ [15]

Контраст объекта с фоном	Характеристика фона	Искусственное освещение		
		Освещённость, лк		
		При системе комбинированного освещения		
		всего	В том числе от общего	
Средний, большой	Светлый, средний	400	200	200

Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Коэффициент пульсации при работе с ПЭВМ не должен превышать 5%. Также следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м² [17].

Для источников искусственного освещения следует применять люминесцентные лампы типа ЛБ, а при устройстве отраженного освещения — и металлогалогенные лампы. В светильниках местного освещения допускается применять лампы накаливания. Для рассеивания естественного

света на окнах должны быть установлены жалюзи (занавески, внешние козырьки и т.п.).

6.1.1.3 Монотонный режим работы

Работа за компьютером подразумевает однообразное и многократное повторение одних и тех же действий, а также малую физическую активность.

Монотонный режим работы может привести к сонливости, невнимательности, чувству усталости и т.п.

Виды трудовой деятельности разделяются на 3 группы: группа А - работа по считыванию информации с экрана ВДТ с предварительным запросом; группа Б - работа по вводу информации; группа В – творческая работа в режиме диалога с ПЭВМ. Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с ПЭВМ, которые определяются: для группы А и Б по суммарному числу считываемых или вводимых знаков за рабочую смену для группы В - по суммарному времени непосредственной работы с ПЭВМ за рабочую смену[16].

Для предупреждения преждевременной усталости рекомендуется чередовать работу с ПЭВМ и без нее. Для повышения эффективности работы устанавливается суммарное время регламентированных перерывов (таблица 24).

Таблица 24 – Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида категории трудовой деятельности с ПЭВМ [16]

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин.	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12-часовой смене
I	до 20 000	до 15 000	до 2	50	80

II	до 40 000	до 30 000	до 4	70	110
III	до 60 000	до 40 000	до 6	90	140

6.1.2. Опасные производственные факторы

6.1.2.1. Опасность поражения электрическим током

Работа с ЭВМ представляет собой работу с электрооборудованием, что может представлять опасность поражения электрическим током.

Электрический ток, который проходит через организм человека, оказывает термическое, электролитическое и биологическое воздействие. Что может привести к ожогам различных частей тела, изменению свойств органических жидкостей, и нарушению протекания в организме различных внутренних биоэлектрических процессов.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации электроустановок и вычислительной техники. Рабочие места с ПЭВМ не следует размещать вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ [16].

Таблица 25 – Напряжения прикосновения и токи, протекающие через
тело человека

Род тока	Напряжение прикосновения, В	Ток, мА
	Не более	
Переменный, 50 Гц	2	0,3
Постоянный	8	1

Напряжения прикосновения и токи приведены при продолжительности воздействия не более 10 мин в сутки и установлены исходя из реакции ощущения.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности:

- Перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;
- При обнаружении неисправности оборудования и приборов необходимо, не делая никаких самостоятельных исправлений, сообщить человеку, ответственному за оборудование [17].

6.1.2.2 Опасность возникновения пожара

Дым, огонь, пониженная концентрация кислорода, повышенная температура окружающей среды, токсические продукты горения – это опасные факторы пожара, которые представляют угрозу для жизни человека, пребывающего в зоне пожара. Регулирование пожаробезопасности производится СНиП 21-01-97.

В помещениях с ЭВМ повышен риск возникновения пожара из-за наличия большого количества электронных устройств.

Устранение причин возникновения пожаров достигается созданием пожаробезопасных технологических процессов; герметизацией, механизацией; автоматизацией; рациональными системами вентиляции, освещения и отопления; применением защитных устройств; осуществлением контроля за состоянием оборудования, технологического и противопожарного режимов[19].

Для предотвращения пожара помещение с ПЭВМ должно быть оборудовано первичными средствами пожаротушения: углекислотными огнетушителями типа ОУ-2 или ОУ-5; пожарной сигнализацией, а также, в некоторых случаях, автоматической установкой объемного газового пожаротушения [18].

6.2 Экологическая безопасность

Непосредственно разработанное программное обеспечение не наносит вреда окружающей среде, но если говорить о более глобальном воздействии, то средства его разработки и эксплуатации – персональный компьютер, может наносить вред окружающей среде.

Современные ПЭВМ производят практически без использования вредных веществ, опасных для человека и окружающей среды. Исключением являются аккумуляторные батареи компьютеров. В аккумуляторах содержатся тяжелые металлы, кислоты и щелочи, которые могут наносить ущерб окружающей среде, попадая в гидросферу и литосферу, если они были неправильно утилизированы. Для утилизации аккумуляторов необходимо обращаться в специальные организации, специализировано занимающиеся приемом, утилизацией и переработкой аккумуляторных батарей [20].

6.3 Безопасность в чрезвычайных ситуациях

Так как рабочее помещение оборудовано ПЭВМ, наиболее вероятностная чрезвычайная ситуация, которая может возникнуть при разработке или эксплуатации проектируемого решения – это ЧС техногенного характера, а именно пожар.

Персональные компьютеры являются наиболее защищенными от возгорания устройствами: им не страшны перепады в сети и внезапные отключения электроэнергии. Однако при нарушении определенных правил использования, даже они могут вспыхнуть пламенем и нанести колоссальный ущерб имуществу и здоровью людей [21].

Возможные причины воспламенения компьютера:

- контакт ПК с водой;
- перегрузка компьютера;
- сломанный вентилятор системного блока.

Для предупреждения возгорания электроприбора необходимо:

- не перегружать электросеть;
- при ощущении специфического запаха горящего пластика, немедленно прекратить использование оборудования.

При возникновении огня, ни в коем случае нельзя гасить водой включенный электроприбор. Включенный электроприбор можно потушить накрыв его сухим одеялом или воспользоваться специальным огнетушителем (порошковым или углекислым).

Если огонь нельзя потушить самостоятельно и в помещении накопилось большое количество дыма, необходимо закрыть окна, выйти на улицу и немедленно вызвать пожарную охрану.

Меры по ликвидации последствий возгорания компьютера:

- демонтаж пострадавшего устройства и замена недееспособных компонентов (если техника еще подлежит ремонту);
- очистка компьютерного стола, пола, стен и потолка от следов гари;
- проведение отделочных работ на наиболее поврежденных участках помещения (после очень сильных пожаров);
- устранение запаха гари в помещении.

6.4 Правовые и организационные вопросы обеспечения безопасности

6.4.1 Специальные правовые нормы трудового законодательства

Трудовые отношения между работниками и работодателями определяет трудовой кодекс РФ. В частности, он устанавливает права и обязанности обеих сторон, регулирует вопросы охраны труда, закрепляет правила оплаты и нормирования труда.

Вид трудовой деятельности на персональном компьютере в рамках данной работы соответствует группе В – творческая работа в режиме диалога

с ПК, категория трудовой деятельности – III (до 6 часов непосредственной работы на ПК).

При 8-часовой рабочей смене и работе на ПК, соответствующей описанным выше критериям необходимо через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва устраивать регламентированные перерывы продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях [22].

6.4.2 Организационные мероприятия при компоновке рабочей зоны

К мероприятиям, относящимся к компоновке рабочей зоны относятся работы по организации рабочего места пользователя, позволяющие наилучшим образом организовать деятельность работника, делая его работу максимально удобной и безопасной.

Требования к рабочему месту программиста:

- яркость дисплея должна иметь оптимальное значение;
- размеры символов на дисплее должны быть легко различимы;
- цветовые параметры дисплея не должны вызывать утомления глаз и головную боль;
- опоры для рук не должны причинять неудобства во время работы на клавиатуре;
- верхний край монитора должен находиться на одном уровне с глазами, нижний – примерно на 20° ниже уровня глаз;
- локтевой сустав нужно держать под углом 90°;
- монитор должен иметь антибликовое покрытие [15].

Для снижения статических физических нагрузок надо использовать мебель, соответствующую санитарным нормам: рабочий стул должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также – расстоянию спинки от переднего края сиденья. Тип рабочего стула должен выбираться в зависимости от характера и продолжительности работы с ПЭВМ с учетом роста пользователя, необходимо соблюдать режим работы, правильную позу.

Заключение

В ходе выполнения выпускной квалификационной работы были рассмотрены основные возможности платформы Roslyn, предоставляющей разработчику различные мощные средства для анализа и рефакторинга кода.

В результате выполнения работы был разработан инструмент рефакторинга для организации ООО «ТомскАСУпроект» в виде плагина Visual Studio, позволяющий удалять автоматически сгенерированные комментарии в открытом решении.

Данное программное средство было применено к проектам организации ООО «ТомскАСУпроект», что позволило избавиться от комментариев, не несущих за собой смысловой нагрузки. Если в компании возникнет очередная потребность в новых функциях для рефакторинга кода, то архитектура плагина позволяет расширять его функционал.

Список использованных источников

1. Википедия. Свободная энциклопедия. [Электронный ресурс] – Режим доступа – URL: <https://ru.wikipedia.org/wiki/Рефакторинг> (дата обращения: 20.01.2018).
2. Описание разработки интерактивных приложений на языке C# с использованием архитектуры .NET Framework. [Электронный ресурс] – Режим доступа – URL: https://professorweb.ru/my/csharp/charp_theory/level3/3_3.php (дата обращения: 25.02.2018).
3. Википедия. Свободная энциклопедия. [Электронный ресурс] – Режим доступа – URL: <https://ru.wikipedia.org/wiki/ReSharper> (дата обращения: 1.02.2018).
4. Википедия. Свободная энциклопедия. [Электронный ресурс] – Режим доступа – URL: [https://ru.wikipedia.org/wiki/CodeRush_\(программа\)](https://ru.wikipedia.org/wiki/CodeRush_(программа)) (дата обращения: 1.02.2018).
5. Википедия. Свободная энциклопедия. [Электронный ресурс] – Режим доступа – URL: https://ru.wikipedia.org/wiki/C_Sharp (дата обращения: 30.01.2018).
6. Документация по Windows. [Электронный ресурс] – Режим доступа – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/add-ins/> (дата обращения: 29.02.2018).
7. Хабрахабр. [Электронный ресурс] – Режим доступа – URL: <https://habr.com/post/130903/> (дата обращения: 11.03.2018)
8. Хабрахабр. [Электронный ресурс] – Режим доступа – URL: <https://habrahabr.ru/company/pvs-studio/blog/301204/> (дата обращения: 1.02.2018).
9. MSDN. [Электронный ресурс] – Режим доступа – URL: <https://msdn.microsoft.com/ru-ru/magazine/dn818501.aspx> (дата обращения: 7.02.2018).

10. The Code Project. [Электронный ресурс] – Режим доступа – URL: <https://www.codeproject.com/Articles/720329/Visual-Studio-Extensions-from-Add-in-to-VSPackage> (дата обращения: 16.04.2018).
11. Хабрахабр. [Электронный ресурс] – Режим доступа – URL: <https://habrahabr.ru/company/pvs-studio/blog/192486/> (дата обращения: 11.04.2018).
12. Беляков Г. И. Охрана труда и техника безопасности: учебник для прикладного бакалавриата. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 404 с.
13. Кукин П.П. Безопасность жизнедеятельности. Безопасность технологических процессов и производств. Охрана труда : учебное пособие для вузов. — 5-е изд., стер. — Москва: Высшая школа, 2009. — 335 с.: ил.
14. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. — М.: Информационно-издательский центр Минздрав России, 1997
15. СНиП 23-05-95. Естественное и искусственное освещение. — М.: Минстрой России, 1995.
16. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. — М.: Информационно-издательский центр Минздрав России, 2003.
17. ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты
18. ГОСТ 12.1.038-82. Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов
19. Шибает Г. И., Гончарюк В. А., Полозков В. Т. Основы техники безопасности и противопожарной техники. М.: Недра, 1967. - 228 с.
20. СанПиН 2.1.7.1322-03. Санитарно-эпидемиологические правила и нормативы. Гигиенические требования к размещению и обезвреживанию отходов производства и потребления.

21. Возгорание персональных компьютеров // Tiny. URL: <http://tinyhack.ru/vozgoranie-personalnyh-kompyutеров/> (дата обращения: 21.05.2018).

22. "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015).

Приложение А.

Таблица – Используемые теги комментария в зависимости от его
владельца

Владелец комментария	Используемые теги		
	summary	param	returns
Класс	+		
Структура	+		
Интерфейс	+		
Конструктор	+	+	
Поле	+		
Свойство	+		
Метод	+	+	+
Перечисление	+		
Элемент перечисления	+		
Делегат	+	+	
Событие	+		

Приложение Б

Таблица – Правила генерации автокомментариев

Владелец комментария / тег	Правило изменения имени владельца	Правило генерации текста комментария
Класс		
Структура		
Поле		
Перечисление		
Элемент перечисления		
Событие		
Метод / summary		
param		
Свойство		«Gets the <i>измененное имя владельца.</i> » или «Gets or sets the <i>измененное имя владельца.</i> »
Конструктор	–	«Initializes a new instance of the <see cref="имя класса"/> class.»
Интерфейс	Из имени удаляется первая буква «I»	«The <i>измененное имя владельца</i> interface.»

Приложение В

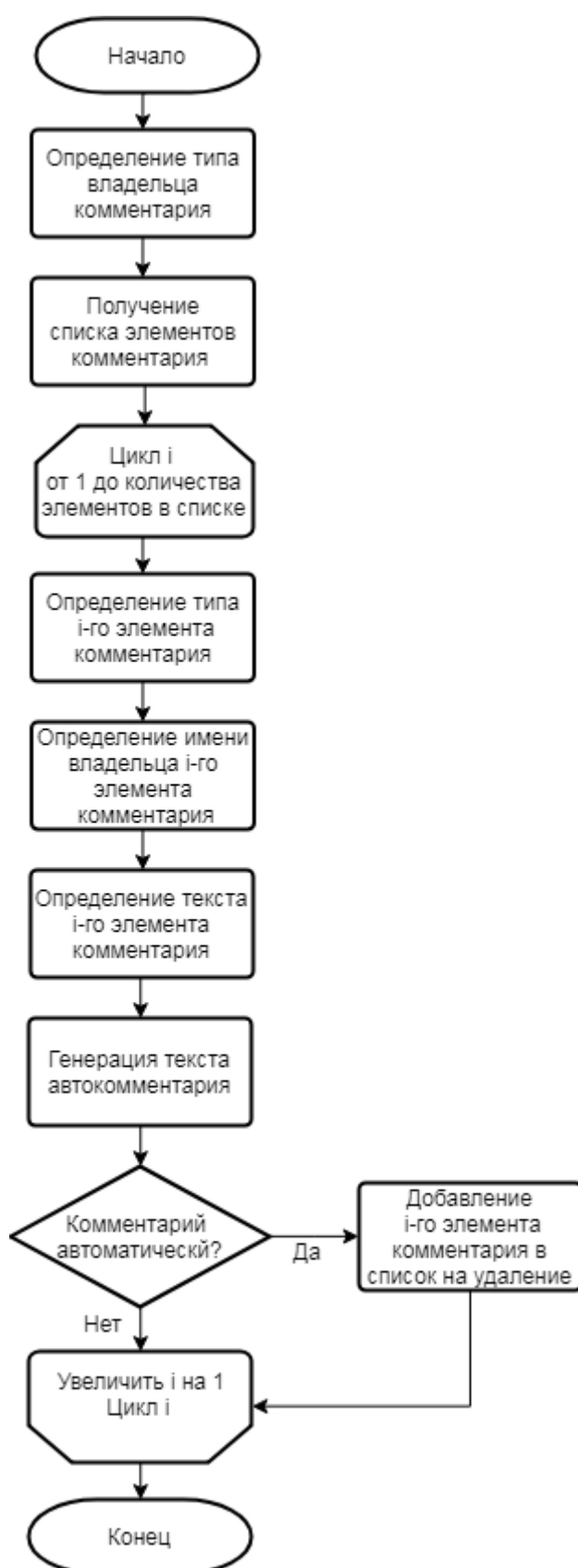


Рисунок В.1 – Алгоритм анализа происхождения комментария